An Integrated and Synergistic Systems Analysis Approach

Gloria B. Isler
Lockheed Martin Information Systems
12506 Lake Underhill Rd., MP 1270
Orlando, FL 32825-5002
gloria.b.isler@lmco.com (407) 306-7419

I. ABSTRACT

A systems engineered approach to evaluation and validation of an integrated hardware/software system requires the coupling of continual improvement of architecture with integrated and synergistic validation methods. Analysis tools supporting this include traditional systems engineering models, object oriented models, static architecture models, and dynamic performance models. An iterative, integrated approach can ensure a fully validated system that meets or exceeds the customer needs, while adhering to the systems engineering process as well as object oriented software design techniques. The iterative nature of the system engineering process enables refinement of the models as system design is further defined to lower levels. The synergistic nature of the analyses permits concurrency and feeding of parameters from one model to another, with the resulting analyses more representative of the whole design, rather than just pieces of the design. Of particular interest during the conceptual development of a system is that generalized, high level assumptions about an architecture or design can be utilized to provide "first look" estimates of the proposed system. The analysis results provide guidance on architecture selections and design techniques. As the design is further refined, these initial models and analyses can be re-used and refined to further explore system capabilities. This approach allows the analyses to mature as the system design matures, but more importantly, the analytical tools that support the verifications and validations will provide design guidance throughout the systems engineering process, ensuring a supportable and interoperable system.

II. SYSTEMS ENGINEERING TECHNICAL PROCESS

Comprising the fundamental systems engineering framework are the technical process activities of (1) Analyze Requirements, (2) Define Candidate Architectures, (3) Optimize and Evaluate Alternatives, and (4) Verify System. It is not unusual during complex system development for a systems engineer to find him/herself in the second or third activity while still defining the requirements. The adaptation of a structured iterative process to systems engineering, similar to the iterative object oriented development of software, can be applied for these design efforts, refining and expanding the depth of the key activities, while providing continuous feedback throughout the development process. Systems engineers have discovered that iteration can ensure traceability of the user's needs to the architectural design and development of the product to a much greater extent for these types of programs. The complexity of high technology product designs is driven by factors such as performance, system interfaces, extent of legacy equipment, and planned technology refreshment at a minimum, and this affects the implementation of the traditional systems engineering process. The transition of traditional systems engineering to an object oriented and structured systems engineering approach can be facilitated during the key technical processes through the use of

Gloria B. Isler
Lockheed Martin Information Systems
12506 Lake Underhill Rd., MP 1270
Orlando, FL 32825-5002
gloria.b.isler@lmco.com (407) 306-7419

integrated tools and by the establishment of integrated engineering processes addressing total system design.

During the ***"Analyze Requirements"*** activity, the engineer performs mission analysis and defines system requirements. Inputs from the customer can be elicited and validated throughout this process, providing input to design verification and compliance. Analysis of compliance to the requirements is performed to identify potential variation in system requirements and is actually used throughout the design process in the development of a design that best accommodates changes. An object oriented requirements management tool should be utilized to establish an integrated requirement management process that facilitates traceability, validation, and control of evolving system design requirements and associated data.

During the ***"Define Candidate Architecture"*** activity, the functional architecture and the system architecture are formulated by defining the functions that the system must perform to achieve the desired outputs, by defining the interfaces between these functions, and by defining the requirements associated with each of these function. Functional architectures can be continually refined throughout the design process based on any additional or modified functions resulting during the integration of system components.

Functional analysis identifies specific capabilities that the system design must incorporate to satisfy the requirements invoked. A visual modeling tool supports functional analysis by modeling the functional requirements of the system and defining the system architecture. The modeling tool supports the derivation and validation of performance requirements in the selected requirements management tool by establishing the functional need for performance. The parameters affecting system performance requirement are identified and their parameter values analyzed during this activity, with the allocation of the parameter values occurring later as part of the "***Optimize and Evaluate Alternatives***" activity.

The techniques and methodologies utilized during the ***"Define Candidate Architecture"*** activity can include functional diagrams, time sequence analysis, system behavior modeling, Object Oriented Analysis (OOA), and Simulation Based Design (SBD). OOA techniques can be used as the foundation of the functional design. Use Case analysis of operational scenarios can be used to develop the components of the system model using Unified Modeling Language (UML) notation and the visual modeling tool. Utilizing UML notation can improve the effectiveness of the analysis by providing the ability to model multiple flows of control, inter-process communication, system behavior, and both software and non-software "things." This technique provides an integrating function that coordinates and deconflicts activities of various members of an engineering team executing parallel development programs.

During the "***Optimize and Evaluate Alternatives***" activity, studies and analyses are conducted, resulting in the optimization and allocation of performance requirements,

Gloria B. Isler
Lockheed Martin Information Systems
12506 Lake Underhill Rd., MP 1270
Orlando, FL 32825-5002
gloria.b.isler@lmco.com (407) 306-7419

assessment of technical risks, and selection of the preferred system architecture. The evaluation criteria and approach are defined, with the candidate architectures studied.

The results of performance, trade studies, and other analyses are instrumental in not only choosing the system design, but also in setting up the framework for validating that the selected architecture as meeting the user's needs. The value of analysis re-use and continued iteration is most apparent during this activity. With careful planning, structure and implementation, analyses can be continually refined for utilization in the subsequent "***Verify System***" activity.

During the "***Verify System***" activity, the system is verified as satisfying its requirements in accordance with verification plans. Verification plans identify the overall integration and test flow, specific tasks, schedules, and resources that would be necessary. Verification methods can be defined and procedures for performing verification developed. The "***Verify System***" activity can be applied to both verification of design and verification of product, and performed incrementally as the design evolves, advocating a structured iterative systems engineering approach. Design verification is generally accomplished by analysis. Early in the development lifecycle, simulation and analysis techniques are used to verify the system. As the system evolves, verification can include integrating and testing of system components and the system. Use of OOA techniques through integrated tools can facilitate the generation and conduct of the verification process. Structured systems engineering processes that maintain the relationships between models and components help ensure the traceability of the design and the integrity of the product.

III. EVALUATION AND VALIDATION MODELS

The adaptation of a structured iterative process to systems engineering necessitates the utilization of evaluation and validation methodology that couples continual improvement of architecture with validation methods that are integrated and synergistic in conduct. This includes not only traditional systems engineering models, but also object oriented models. Figure 1 depicts an iterative, integrated approach that can be used to ensure a fully validated system meeting or exceeding the needs of the customer.

Following is a brief description of how these analyses and models can be utilized to evaluate and model the system being designed. The system can be modeled at various stages of the systems engineering process to include concept development as well as engineering development. The iterative nature of this system engineering process enables the analyst to refine the modeling as the system design is further defined to lower levels. Of particular interest during the conceptual development of a system is that generalized, higher level assumptions about an architecture or design can be utilized in a Static Reliability Assessment, Switchover Analysis, Human/System Modeling, and Dynamic Performance Modeling to provide "first look" estimates of the system proposed. The results can then be utilized to provide guidance on architecture selections and design techniques. As the design is further refined, then these initial analyses can also be refined to further explore the capability of the system.

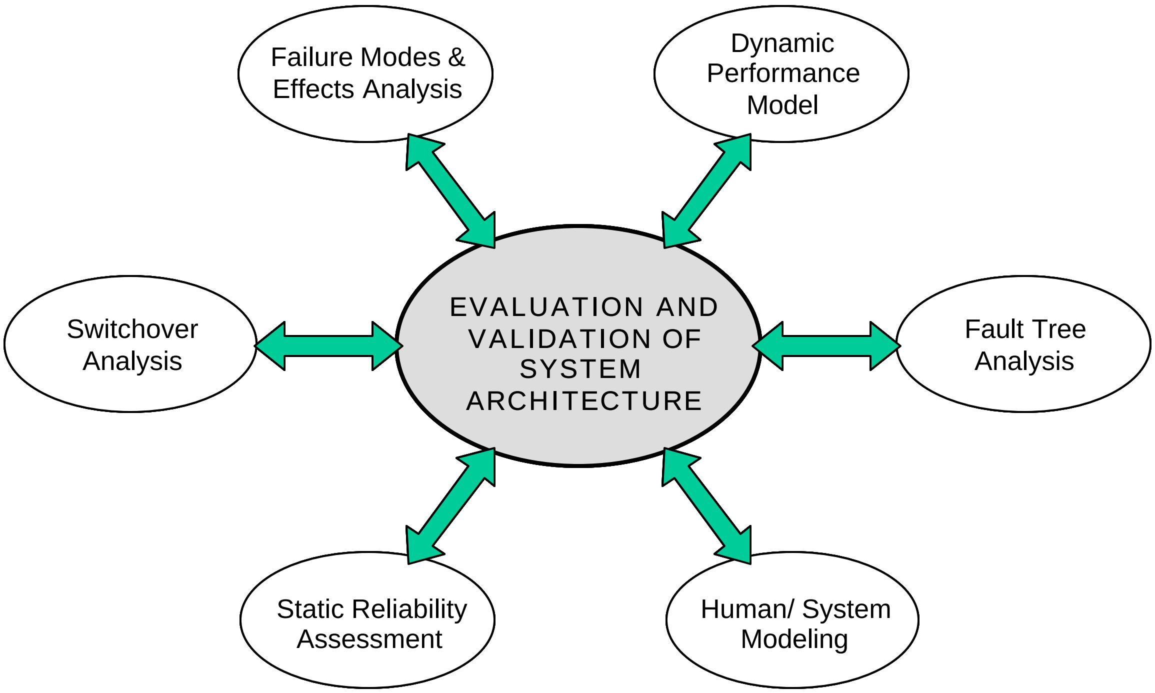

Figure 1 Synergistic System Models

The depiction in Figure 1 is not totally inclusive, as there are numerous other analysis models that could be used in the evaluation and validation process. It should also be understood that it may not be necessary that all of these analysis models be used, as it depends upon assessment needs and results. The models discussed in this section represent those that have proven successful on various programs.

Static Reliability Assessment

A static reliability assessment of the system architecture can be performed using a commercially available reliability tool to model the components of the system. This tool utilizes traditional reliability analysis methods that are based on system architecture, component data (which includes failure rate and maintenance information), and system concept of operations. This process is often referred to as a reliability prediction, which is the analysis of parts and components in an effort to predict the rate at which the system will fail.

An Integrated and Synergistic Systems Analysis Approach

Gloria B. Isler
Lockheed Martin Information Systems
12506 Lake Underhill Rd., MP 1270
Orlando, FL 32825-5002
gloria.b.isler@lmco.com (407) 306-7419

A reliability prediction is most often based on an established model, which provides the procedure for calculation of failure rates for the various components, using standard equations which account for application parameters such as stress parameters, device temperature, operating voltage, and power stress ratios. A reliability prediction analysis requires the use of information about the components in mathematical equations that compute the failure rate for that component. A tool can automate this procedure for the computation of the failure rate.

Assessing the reliability of the system requires the use of component failure rate data and a Reliability Block Diagram for analyzing the system architecture. The design of the system employs redundancies of key assemblies to improve the overall reliability. The analysis of these types of redundancies requires the use of sophisticated mathematical algorithms. However, a tool usually provides a graphical block diagram evaluator for the analysis of these types of configurations. There are some commercially available tools that can perform the static operational availability analysis and some that provide the ability for the analyst to model software components in the reliability assessment. These tool features help ensure a quantified and integrated approach to handling all components, both software and hardware.

The basic premise for the system reliability (and consequently availability) assessment is that the initial evaluation of the system must be performed on the system free of any workload and performance constraints. The analysis assumes a perfect system in regards to the effects of the system workload, measuring only the availability inherent to the equipments, system architecture, operational concept, and support concept. Revealed during this first cut analysis will be the critical items driving the availability of the system when the system is subjected to a perfect workload without any throughput driven constraints. This first order analysis is crucial in that the resulting static system availability is then considered to be the upper limit of the system operational availability.

Switchover Analysis

For equipment clusters in which the operational concept is more complex than “k out of n” objects (meaning that “k” objects are sufficient for correct operation of the node of “n” objects in which “k” is less than “n”), the use of Markov models can provide more meaningful and realistic analysis results. This is the situation in which complex dependencies exist between failure and repair events. The Markov modeling technique utilizes an abstract representation of subsystem states and the behavior governing transitions between the specified states.

In this situation it is preferred to utilize a commercial tool that will perform the static operational availability analysis for a complex client server architecture by providing reliability prediction assessments as well as Markov modeling. To define a Markov model, it is necessary to specify the required set of states, with designations for each state as a “good” state or a “bad” state (e.g., operational or not), and transition rates between

Gloria B. Isler
Lockheed Martin Information Systems
12506 Lake Underhill Rd., MP 1270
Orlando, FL 32825-5002
gloria.b.isler@lmco.com (407) 306-7419

states. The analyst should select a tool that calculates the steady state availability for the entire Markov model, as well as for the individual states within the model.

The utilization of Markov models provides a realistic representation of systems when detailed analysis of fault detection and recovery techniques is required. The equipment clusters requiring Markov modeling should be analyzed in conjunction with the traditional reliability analysis, providing an integrated static reliability assessment that combines both modeling methods.

Failure Modes and Effects Analysis (FMEA)

An FMEA is a bottoms-up approach to analyzing system design and performance. Commercially available tools provide the capability to perform FMEAs. The lowest levels of the system must first be outlined with either the individual components or the lowest level assemblies in the system. For each lowest level, a list of potential failure modes is generated with the tool. The effects of each potential failure mode can then be determined.

Dynamic Performance Model

The network architecture characteristics and projected operating constraints can be analyzed in a dynamic sense, through a discrete event simulation, which initially assumes no failures or degradations due to component failure. This is in contrast to the static reliability assessment, which provides analysis of failures while assuming no workload constraints. Commercially available simulation tools can be utilized in performing the dynamic analysis, often referred to as performance analysis.

Multiple operational scenarios requiring analysis can be modeled, with specific parameters characterizing the network. Typical parameters for this analysis include, but are not limited to, CPU specifics (e.g., processing rate, instructions, and latency), memory specifics, Input/Output (I/O) latency, and network latency. The simulation model can provide a prediction of the expected effects on the system performance as the workload is increased. A model of the network can be useful to evaluate changes in the architecture and "what-if" exercises, as well as changes in the operational concept.

After the calibration of the performance model for the architecture with no failure degradations, then the model can be further exercised to evaluate impacts of singular or multiple failure events. Failure rate and repair information from the static reliability assessment can be used as inputs to the system performance model. Iterative uses of the model to investigate the impacts of failures and design changes provide the eventual validation of the optimal system architecture, with the appropriate characterization of failure events. Maintaining a performance model throughout the life of the engineering process is highly suggested in the design of complex computer architectures. These models are key to the technology refreshment analyses in providing a robust product.

Gloria B. Isler
Lockheed Martin Information Systems
12506 Lake Underhill Rd., MP 1270
Orlando, FL 32825-5002
gloria.b.isler@lmco.com (407) 306-7419

Fault Tree Analysis (FTA)

The FTA is a deductive, top-down method of analyzing system design and performance. It involves specifying a "top event" to analyze, followed by identification of all associated elements in the system that could cause the "top event" to occur. Fault trees can be used to express logical relationships between the operational status of a full tree structure and its components. Combinations or sequences of events leading to system failure can be logically represented with symbols. FTAs are performed graphically using a logical structure of AND and OR gates. The entire system, including software and human interactions, can be analyzed when performing a fault tree analysis.

The availability of a fault tree node can be computed utilizing steady state results computed for the elements within the tree. Some reliability analysis tools allow the analyst to represent hierarchical nodes within fault trees, with the nodes representing Line Replaceable Units (LRUs), other fault trees, or traditional reliability block diagrams.

Human/System Modeling

Human/system models can be utilized to assess and optimize the degree of automation or human in the loop aspects of the system. This type of model incorporates domain knowledge about specific user interfaces, policy and cultural constraints, and operational requirements to evaluate the human/system interfaces. In addition, task analyses can be conducted as necessary to support the human/system modeling efforts. Usability testing can be conducted during display development of Human Machine Interfaces (HMIs) to ensure the optimal human-in-the-loop considerations.

A tool can be utilized to perform such analyses on a system. Human performance factors can be modeled to include, but not be limited to, the following parameters or variables: time to perform; probability of human failure; environmental variables; team compositions; manual and/or semi-automatic operational characteristics; and other general operational characteristics. The dynamics of operations can be modeled in this manner to provide data suggesting the time lines of appropriate operator interventions in the tasks. Results generated during the human/system modeling can be used as inputs to other models, such as the dynamic performance analysis, FTA, and the static reliability assessment, providing a more realistic and comprehensive evaluation and validation of the system.

IV. INTEGRATED SYSTEM ASSESSMENT

The assessment of the design is conducted during the ***"Optimize & Evaluate Alternatives"*** activity. Of particular interest in high technology products is the measurement of system effectiveness through multiple and concurrent analyses. One such approach in providing an overall assessment is depicted in Figure 2. The assessment initiates with research on individual components, architecture design, and operational configurations. The goal is to characterize the individual components of the proposed design in terms of operational and inherent parameters. These characteristics can be in the form of metrics that quantify component failure rates and repair rates, information processing characteristics, or other operational characteristics, such as standard operating procedures, manning requirements, and mission critical functions. Generalized, higher-level assumptions about system design can be utilized initially in the analyses, such as during conceptual development, to provide initial estimates and guidelines for the system architecture. As the architecture is refined, the analyses can also be refined with additional lower level characterizations that can then provide more definitive system evaluations.

The static analysis provides an estimate of the upper limit of the availability or reliability of the defined architecture, without any quantification for the network system operations. The network architecture characteristics and projected operating constraints can be analyzed in a dynamic sense, through a discrete event simulation, which assumes no failures or degradations due to component failure. These results will suggest an upper limit for the system effectiveness with perfect components. Integration of the results of both of these analyses will provide an assessment of the overall system performance, and also suggest design considerations in terms of maintenance philosophies, switchover implementations, and redundancy schemes. Implications of design considerations, such as changes in system operations, automation of specific functions, and fault detection capabilities, can be provided through these analyses.

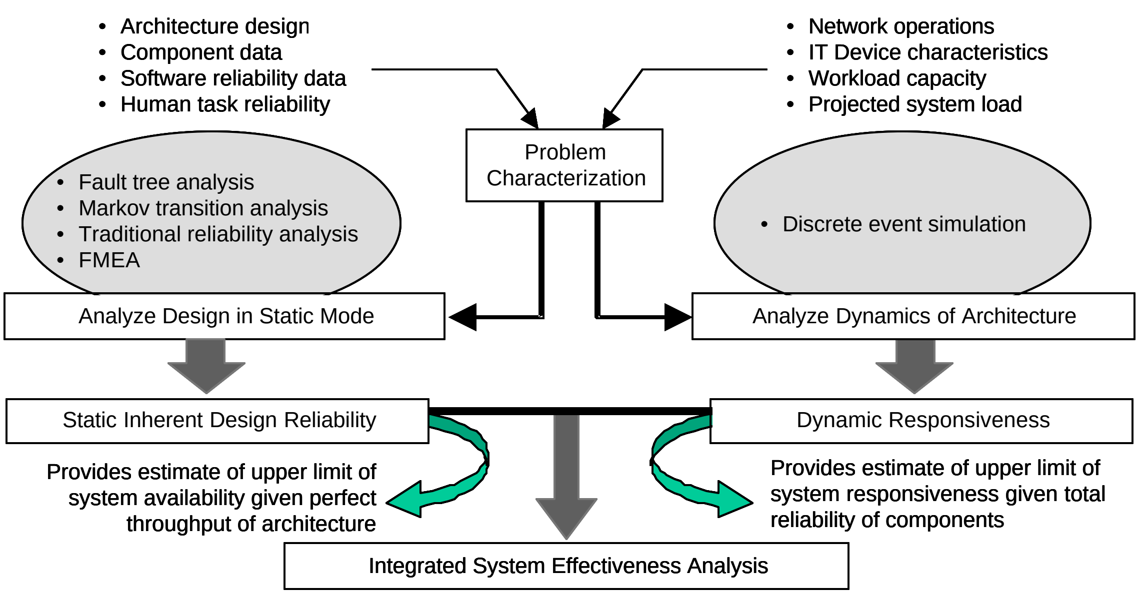

Figure 2 Effectiveness Assessment Approach

An Integrated and Synergistic Systems Analysis Approach

Gloria B. Isler
Lockheed Martin Information Systems
12506 Lake Underhill Rd., MP 1270
Orlando, FL 32825-5002
gloria.b.isler@lmco.com (407) 306-7419

V. INTEGRATION AND TEST

The benefits of continuing to support the system design with synergistic and integrated analyses becomes readily apparent during integration and test. The modeling results are instrumental in the timely and accurate development of systems engineering work products such as the interface documents and test cases. The ability to maintain the relationships between segments and components with integrated tools promotes design traceability and integrity.

One method of facilitating the integration of various software/hardware items is the use of virtual integration. Utilizing this method, initial integration activities may consist of software integration using a virtual integrated software environment network. Developers can connect interfacing software items via this distributed network, enabling interactive visualization of the architecture. Simulations can be generated via a tool that augments the testing by simulating the hardware responses to software commands and requests. Many state-of-the-art commercial products are available as generic dynamic system modeling and simulation tools for those designs in which there is integration of software and hardware. These tools and other similar products provide the ability to insert real hardware in the simulation environment for verifying control hardware performance and for studying control network timing and throughput. The simulation environment not only facilitates validating the system design, but also assists in system analysis.

Interoperability, verification of interface definitions, and initial assessment of COTS/developed code integration can be accomplished during this virtual integration phase. Virtual integration activities, at an increasingly higher level of complexity, can be performed in parallel with actual hardware integration activities. This visualization of the architecture enables validation of the software by demonstrating the initiation of information onto the system's network from any network device. The software engineers can track and measure that propagation of information through the proposed network, providing visualization for analysis purposes of the system behavior and protocol.

Another situation in which system analyses provide integration support occurs when the system being designed is actually one subsystem of a large system. The integration of one company's subsystem with another company's subsystem must be addressed prior to physical integration. This creates the need for software that emulates the performance of other subsystems. Sample subsystems, or representative alternative hardware, can be resident to allow the items to be integrated and tested in an arrangement that represents the anticipated installed configuration. Simulators and stimulators will be used to provide inputs to and accept the outputs from the subsystem representation. Specifically, there will be simulators and/or stimulators that will enable the test configuration to emulate the interfaces. Integration activities will culminate with validation tests of the representative subsystem. Results of these validation tests form the baseline for

Gloria B. Isler
Lockheed Martin Information Systems
12506 Lake Underhill Rd., MP 1270
Orlando, FL 32825-5002
gloria.b.isler@lmco.com (407) 306-7419

prototype and production tests, and can also be utilized in further analysis efforts that characterize and allow visualization of the subsystem behavior.

VI. SUMMARY

Establishing a systems engineering environment that promotes integrated engineering during the entire life cycle can be a big challenge. However, the benefits become more apparent with design maturation. If the environment is established appropriately and concurrent with manpower planning, personnel can migrate throughout the key activities during the life cycle, providing the domain knowledge that is critical during the systems integration and test (SIT) and systems acceptance test (SAT). The ability to easily verify and validate requirements with familiar and knowledgeable personnel demonstrates to the customer that the system has been thoroughly evaluated and demonstrated.

By invoking total system design with object oriented systems engineering (OOSE) processes, a company establishes Integrated Engineering processes early in the design, paving the way for assessment with the Integrated Capability Maturity Model (CMMI). INCOSE has recognized the advantages that OOSE can provide during system design and has initiated a working group for object oriented systems engineering methodology (OOSEM). This working group is supportive of the development of an UML profile for system engineering. It is recognized that tool vendors should be involved in developing tool support for OOSEM, as current systems engineering tools only support OOSEM to a degree, and often require workarounds to be effective. Currently the use of synergistic models and analyses requires early identification of the needs, followed by consistent and continual handshaking throughout the analysis efforts. Focus within the industry on OOSE and the development of such tools will positively influence the trend towards integrated engineering. With enhancements to systems engineering tool suites and the establishment of integrated engineering practices that address total system design, systems engineers can more effectively ensure supportable and interoperable system designs.